Claims 1-3, 6-16 and 19-27 have been amended by this Amendment A.

Claims 4, 5, 17, and 18 have been cancelled by this Amendment A.

Claims 1-3, 6-16 and 19-27 are still in the Application and reconsideration of the Application is respectfully requested.

Referring to Paragraph 2 of the above-identified Office Action, Claims 1-27 have been rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 5,524,244 issued in the name of Robinson et al. Before summarizing the Robinson reference, the invention disclosed and sought to be protected by the Application will be summarized. In the present invention, of the source code suitable for a first signal processing machine is translated into the translation code for use with a different processor or for a different mode of operation. The source code program elements are identified, e.g., by the front end parser of Fig 1. Then each source program element is translated by a program to provide the translation program elements. However, the ability of a program to translate without error the elements of source code program for a first machine into the translation (source) program elements for a second file in a manner that the translation file can be immediately applied to a second data processing machine is very limited. Indeed, programmer intervention is almost always needed to adjust the source program elements for the first machine into the source code program elements for the second machine. However, any adjustment in the translation code can have other implications. Because of these implications, the process is iterated until a workable translation program is provided. In the past, to determine whether the translation code was satisfactory, the code either had to be run and errors noted, or a programmer could examine the program to detect the errors. In the present invention, the translation process includes maintaining the source program elements and displaying the corresponding translation program elements proximate to the source program elements. In this manner, a programmer can compare the corresponding source and translation program elements in the correct position in the software program and can more easily correct the translation program elements to insure that both the source program and the translation program are consistent.

In contradistinction to the disclosure of the Application, the Robinson reference does not relate to the translation of software (assembly) language programs. The Robinson reference, as indicated by the Abstract, relates to the implementation of programs involving both a microprocessor and a real time processor. Because the programs for the microprocessor and the real time processor must be compilied separately, the program is initially is represented by functional blocks. A graphic interface unit is used to convert the functional blocks to software program elements. The graphics interface unit provides a convenient tool for separating the functional blocks representing the real time processor program elements and the microprocessor program elements for the separate compilation. In addition, the graphics interface unit is used to provide characterizing parameters for the functional blocks. The graphic interface unit is, in this function, not operating to translate one software program into another software program. The separate portions of the program are then provided to the microprocessor for separate compilation. Although reference is made to software translation, the translation referred to in the context of the reference is the translation of the "said symbolic objects and connections between symbolic into object code for said signal processor" (of Claim 3 of the reference). Because the translation of symbolic code into object code is not equivalent to the translation of a first software program into a second software for a different processor or for a different mode of operation, the Robinson reference appears to be non-analogous art with reference to the claims of the Application.

Referring specifically to the rejection of Claim 1 beginning on Page 2 and continuing on Page 3 of the Office Action, the front end of the present invention does, as indicated by Examiner, identify source elements. However, with respect to the back end, in the section referenced by Examiner (Robinson 4: 12-20) the graphic interface unit is described as providing "a graphic user interface system for a real time signal processor interfacing with a host microprocessor where the real time signal processor program is complied separately from the program of the microprocessor , but as part of the compiling procedure provides the microprocessor-related file to the microprocessor which then translates the file and incorporates the translated file into its compilation, and thereby automatically provides for the signal processor-microprocessor interface." This quotation is perfectly consistent with the

description of the Robinson reference given above. As indicated previously, the Robinson reference does not provide for a translation of a first software program to second software program. This quotation indicates that the real time file generated from the symbolic circuit elements transferred to the microprocessor for translation. However, the translation referred to in this section is also consistent with the translation of the symbolic elements into object code described above. Then, the translated program is described as being "compiled". No indication is given in the Robinson reference that this (translation) object code is applied to the back end (graphic interface unit) as described in Claim 1. In view of the fact that the Robinson reference has apparatus that performs a function different from the function of the apparatus of the Application, the Application is patentably distinct from the reference. Furthermore, claim 1 includes a limitation (application of the translation code to the back end) that is not found in the reference. Therefore, rejection of Claim under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 5,524,244 issued in the name of Robinson is respectfully traversed.

Claims 2-3 and 6-13 depend from Claim 1, a claim now believed to be in condition for allowance. Consequently, Claims 2-3 and 6-13 are believed to be in condition for allowance. Therefore, rejection of Claims 2-3 and 6-13 is respectfully traversed. With specific reference to Claim 10 and Examiner's comments concerning this Claim 10 on Page 4 of the Office Action, Applicants' attorney is unable to find in Robinson 53:1-25 any reference to "at least a portion of the source elements in a source window, at least a portion of the translation elements in a transition window, and source and translation windows are displayed side-by-side". The translation elements are never applied to the graphics interface unit in the Robinson reference. As indicated above, the graphics interface unit is used to convert functional symbols to software structures for use in implementation of a program involving both the microprocessor and the real signal processor. This conversion is not a translation from one software program to a second software program. The translation, as described above, is from a symbolic representation to an object code. In addition, as indicated above, no translation elements are applied to the graphics interface unit.

Referring specifically to the rejection of Claim 14 described on Page 5 of the Office Action, Claim 14, as indicated by Examiner, does include receiving a source

program and identifying the source elements. The present invention provides a translation program and the elements of the translation program are displayed for user inputs. Nowhere in the Robinson reference is the display of the translation program elements on an interface "for receiving user inputs" described. In the Robinson reference, symbols of functional software processes are displayed on the graphics interface unit. Software elements related to the software processes are displayed along with parameters from the various registers. However, this does not constitute a software program to software program translation, but merely a change of a symbolic form of representation of a program to a software oriented representation. (Applicant is unable to find the words "context-dependent" in the section cited by Examiner.) Applicant respectfully points out that not in the section cited by Examiner, or in any other part of the reference, are the translation elements described as being applied to the interface unit (back end or graphics interface unit). In the Robinson reference, the user inputs are provided to an interface unit to separate the real time elements from the microprocessor elements and to supply parameters needed to characterize the real time processor procedures. Nowhere in the section cited by Examiner, or in any other portion of the Robinson reference, are the translation elements described as regenerating the translation file. And the general comment that the reference and the Application appear to be non-analogous art is still true. Therefore, rejection of Claim 14 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 5,524,244 issued in the name of Robinson reference is respectfully traversed.

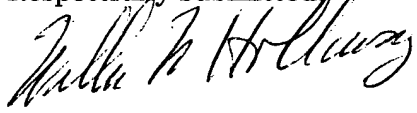
Claims 15-16 and 19-26 depend from claim 14, a Claim believed to be in condition for allowance. Consequently, Claims 15-16 and 19-26 are believed to be in condition for allowance. Therefore, rejection of Claims 15-16 and 19-26 under 35 U.S.C. 102(b) as being anticipated by Robinson is respectfully traversed.

Referring specifically to the rejection of Claim 27 on Page 7 of the Office Action, the Claim 27 has been amended to clearly state that the invention relates to a software program to software program translation. As indicated above, the Robinson reference does not relate to a software-to-software translation, but rather is a tool for designing and implementing a program including a real time processor and a microprocessor. As such, the input to the system described by Robinson is a file of

symbolic elements arranged to provide a specific procedure. The system provides a technique for converting the elements to a representation that can be translated into object code. The input to the system is not a software program, but a group of symbolic elements representing the procedures of a program. The Robinson reference can consequently be described as non-analogous art. Nor does the Robinson reference describe an interface unit having translation program elements applied thereto. In the present invention, this is not a matter of design choice, but is needed to insure the accurate program-to-program translation. Therefore, rejection of Claim 27 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 5,524,244 issued in the name of Robinson is respectfully traversed.

In view of the foregoing discussion and the foregoing amendments, it is believed that Claims 1-3, 6-16 and 19-27 are now in condition for allowance and allowance of Claims 1-3, 6-16 and 19-27 is hereby respectfully requested. Should any matters remain incomplete that can be resolved by a telephonic interview, Examiner is requested to call the undersigned attorney at 281-274 4064.

Respectfully submitted,



William W. Holloway
Attorney for Applicants
Reg. No. 26,182

Texas Instruments Incorporated
P.O. Box 655474, MS 3999
Dallas, TX 75265
(281) 274-4064

Appendix

1. (Amended) A translation system for translating a source assembly language program for a source device into a translation assembly language program for a target device, the system comprising:

a front end for identifying source elements in [a] the source [file] program; and

a back end for generating [a] the translation [file] program having translation elements corresponding to translation of [said] the identified source elements, the backend including interface for comparing the identified source elements with the corresponding translation elements, [and having an] the interface [for] receiving inputs for modifying [said] the translation elements.

2. (Amended) The system [of] as recited in [Claim] claim 1, wherein the source [file] program is for a source device and the translation [file] program is for a disparate target device.

3. (Amended) The system [of] as recited in [Claim] claim 1, wherein the source [file] program is a linear assembly file for a target device and the translation [file] program is a scheduled assembly file for that device.

Please delete Claims 4 and 5 without prejudice.

6. (Amended) The system [of] as recited in [Claim] claim 1, wherein [said] the translation is a context-dependent translation based on static analysis of the source [file] program.

7. (Amended) The system [of] as recited in [Claim] claim 1, wherein the back end further comprises:

a translator for performing a context-dependent translation, the translator comprising:

a translation machine description for mapping source opcodes to target opcodes;

a source machine description containing a description of source opcodes and source operands in a generic representation;

a target machine description containing a description of target opcodes and target operands in a generic representation; and

wherein the translator receives a source instruction from said front end, utilizes the translation machine description and source machine description and target machine description to translate source elements into target elements.

8. (Amended) The system [of] as recited in [Claim] claim 7, wherein the proper target opcode is chosen from a group of potential target opcodes by comparing the target opcode and target operand with the source opcode and source operand.

9. (Amended) The system [of] as recited in [Claim] claim 7, wherein two or more source opcodes can be combined to a single target opcode when there is a target opcode that represents the two or more source code opcodes.

10. (Amended) The system [of] as recited in [Claim] claim 1, wherein the user interface is a graphical user interface.

11. (Amended) The system [of] as recited in [Claim] claim 10, wherein the graphical user interface displays at least a portion of the source elements in a source window, at least a portion of the translation elements in a translation window, and the source and translation windows are displayed side-by-side.

12. (Amended) The system [of] as recited in [Claim] claim 11, wherein corresponding groups of elements of the source and translation [files] programs are aligned in the source and translation windows.

13. (Amended) The system [of] as recited in [Claim] claim 11, wherein at least one of the source and translation windows is operable to display a status icon for an element in the window.

14. (Amended) A method for performing translation of an assembly language source program into an assembly language translation program, the method comprising:

receiving [a] the source [file] program;

identifying source elements in the source [file] program;

generating [a] the translation [file] program having translation elements by performing a context-dependent translation of the source elements;

displaying the translation elements in an interface for receiving user inputs; and

in response to user inputs, automatically regenerating selected translation elements based on the user inputs.

15. (Amended) The method [of] as recited in [Claim] claim 14, wherein the source [file] program is for a source device and the translation [file] program is for a disparate target device.

16. (Amended) The method [of] as recited in [Claim] claim 14, wherein the source [file] program is a linear assembly [file] program for a target device and the translation [file] program is a scheduled assembly [file] program for [said] the target device.

Please delete Claims 17 and 18.

19. (Amended) The method [of] as recited in [Claim] claim 14, further comprising:

performing static analysis of the source elements in the source [file] program; and

performing context-dependent translation of the source elements based on the static analysis.

20. (Amended) The method [of] as recited in [Claim] claim 14, wherein the step of generating a translation [file] program further comprises:

converting an opcode of a source machine to an opcode of a translation machine [file] program by comparing the source opcode to possible translation opcodes;

converting the operand of the source opcode by comparing an operand of the source opcode in a generic expression with a generic expression for a translation operand;

combining the translation opcode and the translation operand to form a translation.

21. (Amended) The method [of] as recited in [Claim] claim 20, wherein the step of converting an opcode of the source file further comprises choosing a translation opcode from a group of potential translation opcodes by comparing the translation opcode and translation operand with the source opcode and source operand.

22. (Amended) The method [of] as recited in [Claim] claim 20, wherein the step of converting the source opcode further comprises the step of combining two or more source opcodes into a single translation opcode when there is a translation opcode that represents the two or more source opcodes.

23. The method [of] as recited in [Claim] claim 14, wherein the user interface is a graphical user interface.

24. The method [of] as recited in [Claim] claim 23, further comprising:

displaying the source elements in a source window;

displaying the translation elements in a translation window; and

displaying the source and translation windows side-by-side in the graphical user interface.

25. (Amended) The method [of] as recited in [Claim] claim 24, further comprising aligning corresponding groups of elements of the source and translation [files] program in the source and translation windows.

26. (Amended) The method [of] as recited in [Claim] claim 24, further comprising displaying a status icon for an element in at least one of the source and translation windows.

27. (Amended) A translation system for translating a source program into a translation program, the system comprising:

a computer capable of executing a program, [and]

an interactive program for translating code for a first processor into code for a second processor and capable of being executed on said computer, and

a graphics interface system displaying source program elements proximate corresponding translation program elements, the graphics interface system permitting correction of the translation program elements by a user.